Resolving issues in P802.1CS D1.4 with making TCP connections and exchanging Hellos

Norman Finn
Huawei Technologies Co. Ltd

cs-finn-hello-tcp-connect-0518-v011

Many thanks to

- Paul Congdon
- Paul Bottorff

For sharing their ideas for solving these problems. These discussions led to the following ideas.

The proxy problem

- The state machines in D1.4 use the IP addresses observed, when the TCP connection is made, to decide which Hellos to send and which Hellos to expect.
- Network Address Translation (NAT) can make the addresses seen in LLDP different from those seen when the TCP connection is made.
- Therefore, **the path from TCP connections to Hello state machine creation in D1.4 does not work**.

A proposal

- We **add a new bit to the LRP LLDP TLVs** to say either, “This is the IP address of a proxy system,” or “This is the address of a direct participant.”
  - (A link-local address would never be used for a proxy system.)
- If **both addresses are link-local**:
  - We can resolve the active/passive question using either the {SystemId, PortId, AppId} or the IP addresses, all of which are in the LLDPDUs.
  - We also know exactly which Hellos to send and expect; they are from the LLDPDUs, also.

A proposal

- If both addresses are **not link-local** (1 local 1 not is not allowed):
  - A proxy system always maintains a passive TCP open request.
  - If one system is a participant and one a proxy system:
    - The participant always does the active open
    - The participant always sends its Hellos first.
    - The proxy system uses received Hellos to create Hello state machines and start sending.
  - If both systems are participants
    - The {SystemId, PortId, AppId} decides who plays the active TCP role and who the passive.
    - *NOTE 1: If the LLDP info is configured, instead of running LLDP, then it could be misconfigured so that no TCP connection can be made.*
    - *NOTE 2: Changing LLDP information can result in two TCP connections. See below.*
  - If both systems are proxy systems …

… A proposal

- … **Both systems are proxy systems** (not link-local, passive TCP open)
  - Because of NAT, we cannot use the IP addresses to decide who is active.
  - Because of NAT, we do not know what Hellos to send when we receive a connection on a passive open.
  - We could use {SystemId, PortId, AppId} information to resolve who makes the active open, but a proxy system may proxy for a large number of such triples, and any mistake could prevent making a TCP connection.
  - I propose that is safer for both sides of a proxy-proxy relationship make an active open, thus making two TCP connections.

… A proposal

- Both systems are proxy systems and have **two TCP connections**
  - The system doing the active open knows what Hellos to send, because it knows for which {SystemId,PortId,AppId} it opened the connection.
  - The system doing the passive open waits to receive Hellos, and responds to them as appropriate.
  - If the two sides discover that they are exchanging Hellos for one portal association on two different TCP connections, they can use {SystemId,PortId,AppId} + who made the active open to determine which state machines to discard.
  - Because the active side knows to what IP address it sent the open, it knows which of its active open connections to reuse for newly-discovered portal associations.

Remaining issue

- It is not yet clear what method will work for dropping one of the two TCP connections between proxy systems.
- But, it would not kill the project if both TCP connections remain.

Thank you